**TP : Manipulation du DOM en JavaScript**

**Objectif**

L’objectif de ce TP est de vous familiariser avec les concepts de manipulation du DOM en JavaScript. Vous apprendrez à :

1. Créer, modifier et supprimer dynamiquement des éléments HTML.
2. Utiliser des événements pour interagir avec le DOM.
3. Modifier les styles et le contenu des éléments via JavaScript.

**Prérequis**

* Connaissances de base en HTML et CSS.
* Connaissances de base en JavaScript (variables, fonctions, boucles, conditions).

**Contexte**

Vous allez créer une page web pour gérer une liste de tâches (to-do list). Cette page permettra d'ajouter, de modifier et de supprimer des tâches. À chaque modification de la liste, le DOM sera mis à jour dynamiquement pour refléter les changements.

**Exigences**

1. **Interface Utilisateur :**
   * Un champ de texte pour entrer une nouvelle tâche.
   * Un bouton « Ajouter » pour ajouter la tâche à la liste.
   * Une liste où chaque tâche s'affiche avec :
     + Le nom de la tâche.
     + Un bouton « Modifier ».
     + Un bouton « Supprimer ».
2. **Fonctionnalités :**
   * Ajouter une tâche : Lorsqu'un utilisateur entre une tâche dans le champ de texte et clique sur « Ajouter », la tâche doit apparaître dans la liste.
   * Modifier une tâche : Lorsque l'utilisateur clique sur le bouton « Modifier » d'une tâche, le texte de la tâche doit devenir éditable. Après modification, il doit pouvoir sauvegarder le texte en cliquant à nouveau sur « Modifier » (qui devient « Sauvegarder »).
   * Supprimer une tâche : Lorsque l'utilisateur clique sur « Supprimer », la tâche est retirée de la liste.
   * Messages d'erreur : Si l'utilisateur tente d'ajouter une tâche vide, affichez un message d'erreur sous le champ de texte.

**Éléments fournis**

**1. HTML de base**

html

Copier le code

<!DOCTYPE html>

<html lang="fr">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Liste de Tâches</title>

<style>

.task { margin-bottom: 10px; }

.edit-mode { border: 1px solid blue; }

</style>

</head>

<body>

<h1>Ma Liste de Tâches</h1>

<input type="text" id="taskInput" placeholder="Nouvelle tâche...">

<button id="addTaskButton">Ajouter</button>

<p id="errorMessage" style="color: red;"></p>

<ul id="taskList"></ul>

<script src="script.js"></script>

</body>

</html>

**2. Script JavaScript**

Créez un fichier script.js et ajoutez-y les fonctionnalités suivantes :

**Instructions**

1. **Ajouter une tâche :**
   * Sélectionnez le champ de texte (#taskInput) et le bouton Ajouter (#addTaskButton).
   * Ajoutez un événement click au bouton Ajouter pour appeler une fonction qui :
     + Récupère le texte du champ.
     + Vérifie que le champ n'est pas vide.
     + Crée un élément <li> contenant le nom de la tâche et deux boutons, « Modifier » et « Supprimer ».
     + Ajoute cet élément <li> à la liste (#taskList).
2. **Modifier une tâche :**
   * Pour chaque élément de tâche ajouté, ajoutez un événement click au bouton « Modifier » qui :
     + Transforme le texte de la tâche en champ de texte (en mode édition).
     + Change le texte du bouton de « Modifier » en « Sauvegarder ».
     + En mode édition, en cliquant à nouveau sur le bouton, sauvegarde le texte et repasse le bouton en mode « Modifier ».
3. **Supprimer une tâche :**
   * Ajoutez un événement click au bouton « Supprimer » qui supprime l'élément <li> correspondant de la liste.
4. **Messages d'erreur :**
   * Affichez un message d’erreur (#errorMessage) si l’utilisateur tente d’ajouter une tâche vide.

**Exemple de code JavaScript (script.js)**

document.getElementById('addTaskButton').addEventListener('click', function() {

const taskInput = document.getElementById('taskInput');

const taskText = taskInput.value.trim();

const errorMessage = document.getElementById('errorMessage');

if (taskText === '') {

errorMessage.textContent = "Veuillez entrer une tâche.";

return;

} else {

errorMessage.textContent = '';

}

const taskItem = document.createElement('li');

taskItem.className = 'task';

taskItem.textContent = taskText;

// Bouton Modifier

const editButton = document.createElement('button');

editButton.textContent = 'Modifier';

editButton.addEventListener('click', function() {

if (editButton.textContent === 'Modifier') {

const editInput = document.createElement('input');

editInput.type = 'text';

editInput.value = taskItem.textContent;

taskItem.textContent = '';

taskItem.appendChild(editInput);

editButton.textContent = 'Sauvegarder';

} else {

const editInput = taskItem.querySelector('input');

taskItem.textContent = editInput.value;

editButton.textContent = 'Modifier';

}

});

// Bouton Supprimer

const deleteButton = document.createElement('button');

deleteButton.textContent = 'Supprimer';

deleteButton.addEventListener('click', function() {

taskItem.remove();

});

taskItem.appendChild(editButton);

taskItem.appendChild(deleteButton);

document.getElementById('taskList').appendChild(taskItem);

taskInput.value = '';

});

**Livrables**

1. Le code HTML finalisé.
2. Le fichier JavaScript script.js avec les fonctionnalités demandées.
3. Un fichier CSS (si vous souhaitez personnaliser l’interface).

**Extensions possibles**

* Ajouter un bouton « Tout effacer » pour supprimer toutes les tâches de la liste.
* Ajouter un effet visuel sur les tâches terminées (par exemple, les barrer).
* Sauvegarder la liste de tâches dans le localStorage pour qu'elle persiste après le rechargement de la page.